Fake News Detection Using Machine Learning, Deep Learning, and Transformer-Based Models

Authors: Jieni Tang*, Xinyu Liu*, Xueni Tang*

Abstract:

The increasing spread of false information across digital platforms has had profound consequences on political discourse, economic behavior, and societal trust. Manual verification of news articles is inefficient, necessitating automated solutions. This study develops an automated fake news detection system employing traditional Machine Learning (ML) models (Support Vector Machine, Random Forest, Naïve Bayes), Deep Learning (DL) models (Recurrent Neural Network, Long Short-Term Memory), and a Transformer-based model (BERT). By integrating enhanced feature extraction techniques and performing comprehensive evaluations across these methodologies, our approach achieves high accuracy, precision, recall, and F1-score, highlighting the effectiveness and robustness of transformer-based methods compared to conventional ML and DL approaches.

1. Introduction:

The wide spread of fake news through social media platforms significantly impacts public opinion and decision-making, and even produces political and economic instability (Shaik et al., 2023). The dissemination of fake news influences public perception and decision-making, often manipulating opinions through misleading or fabricated content (Pérez-Rosas et al., 2017). A major reason for this is that most social media platforms lack built-in mechanisms to distinguish between false and true headlines (Domenico et al., 2021). While independent fact-checking organizations such as *PolitiFact*, *Snopes*, and *TruthOrFiction* have been established to evaluate the credibility of viral claims, these human-centered methods are inherently labor-intensive and

struggle to scale in response to the speed and volume of online misinformation (Saeed & Al Solami, 2023).

This pressing challenge motivates the development of automated systems capable of distinguishing between fake and real news. With advancements in Artificial Intelligence (AI) and Machine Learning (ML), it is now feasible to train systems on existing datasets to perform accurate and scalable fake news classification in real-time. These approaches are not only faster but have shown promising results across numerous benchmarks.

However, the development of automated solutions still faces significant challenges. Many systems depend heavily on surface-level textual features like word frequency or syntactic patterns, which often fail to capture deeper semantic and contextual nuances. Furthermore, there is limited research comparing traditional machine learning (ML), deep learning (DL), and transformer-based approaches using a consistent dataset and evaluation pipeline, making it difficult to assess which methodologies are most effective across varied types of news content.

To address these gaps, our research focuses on the following research question: Among traditional machine learning models, deep learning architectures, and transformer-based approaches, which performs better in fake news detection? To answer this question, we design a comparative study involving three ML classifiers—Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB); two DL architectures—Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM); and the transformer-based model BERT. We preprocess raw data to extract textual features using TF-IDF, Bag of Words (BoW), and Keras. BERT is applied independently to produce contextual embeddings. The derived features are then input into respective models for classification. Performance is evaluated on two public datasets using standard metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

Our Contributions:

Xueni Tang: Introduction & Literature Review & Machine Learning Model

Jieni Tang: Proposed Methodology & Datasets Statistics & Transformer Model

Xinyu Liu: Deep Learning Model & Results Discussion & Demo Design

2. Literature Review

Initial efforts in fake news detection focused on traditional ML approaches and shallow neural networks. Janze and Risius (2017) automated fake news analysis through standardized misleading news detection on social media, using early classification techniques. Building upon this, Pérez-Rosas et al. (2017) explored a mixture of literal and contextual ML strategies to enhance model accuracy across datasets. These early methods, however, lacked the capacity to learn contextual dependencies within text.

Subsequent developments embraced deep learning. Bahad et al. (2019) introduced a hybrid LSTM model trained on benchmark datasets, offering improved performance in sequence modeling tasks. Zhu et al. (2018) further expanded detection capability by proposing an adaptive system that dynamically updated its knowledge base to respond to evolving misinformation. Parallel to these efforts, Pulido et al. (2020) examined health misinformation, analyzing textual features and social impact, supported by broader studies of online discourse.

As attention shifted toward content structure and clustering, Anoop et al. (2020) used CNNs with clustering algorithms like K-Means and Density-based Spatial Clustering of Applications with Noise, distinguishing real from fake health articles through feature pattern analysis. In parallel, Han et al. (2021) leveraged Graph Neural Networks(GNNs) to model fake news diffusion patterns using FakeNewsNet. Building on this, Khan et al. (2021) evaluated hybrid architectures like convolutional-LSTM and Hierarchical Attention Networks(HAN) incorporating preprocessing

pipelines to reach 95% accuracy. Monti et al. (2019) contributed a geometric deep learning model leveraging Twitter data, while Guo et al. (2019) incorporated emotional signals from users and publishers for more nuanced classification.

Recent studies increasingly adopt ensemble and transformer-based strategies that achieve higher accuracy. Hakak et al. (2021) fused decision trees, random forests, and extra trees across multiple datasets to achieve peak accuracy of 99.29%. Choudhary et al. (2021) introduced BerConvoNet, a hybrid of BERT and CNN layers achieving >90% accuracy across benchmark datasets. Ahmad et al. (2020) developed an ensemble combining ML and DL models, confirming the advantage of integrating shallow and deep learners for robust performance.

Despite these advancements, there remains limited comprehensive comparative analysis across diverse methodologies, highlighting a critical research gap this study aims to fill by systematically evaluating various ML, DL, and transformer-based methods. To address this gap, our study comprehensively compares various traditional ML, deep learning, and transformer-based approaches. We combine advanced feature extraction methods, including contextual embeddings from BERT, with established traditional techniques such as TF-IDF and BoW, aiming to identify the most effective strategies for accurate fake news detection.

3. Methods

Our fake news detection framework integrates three distinct modeling strategies, all of which are grounded in a unified preprocessing pipeline. These strategies include traditional machine learning models, deep learning architectures, and transformer-based approaches, each offering unique capabilities in handling textual data.

The overall workflow of the proposed fake news detection system, encompassing data preprocessing, feature extraction, and model training across traditional machine learning, deep learning, and transformer-based architectures, is illustrated in **Figure 1**.

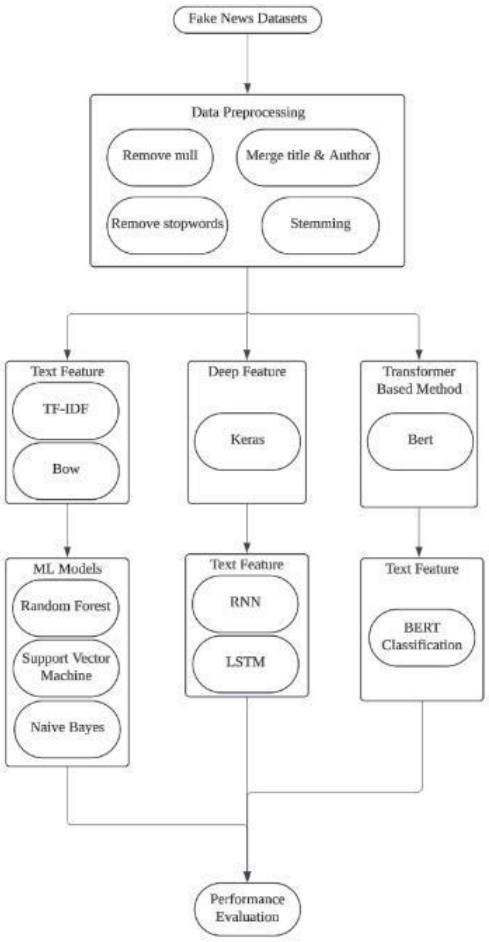


Figure 1: Overview of the proposed fake news detection framework. The process includes data preprocessing, feature engineering, and classification using traditional machine learning models, deep learning architectures (RNN, LSTM), and transformer-based models (BERT).

3.1 Data Preprocessing

The preprocessing stage is essential for standardizing the input data. Initially, all null values are removed and empty fields are filled with placeholders. The title and author fields are concatenated as the input text. Standard NLP steps such as lowercasing, non-alphabet filtering, stop words removal, and stemming are applied. For deep models, word tokenization and padding are used to form uniform-length inputs.

3.2 Traditional Machine Learning Models

The traditional machine learning approach employed in this study leverages two primary text vectorization techniques: Term Frequency-Inverse Document Frequency (TF-IDF) and Bag-of-Words (BoW). Both methods were configured to extract a maximum of 8000 features while excluding standard English stop words. The raw textual data, following normalization and stemming, was transformed into sparse feature matrices to serve as input for classification algorithms.

Support Vector Machine (SVM) with a linear kernel was trained using both TF-IDF and BoW feature representations. SVM was chosen for its effectiveness in high-dimensional text classification tasks, particularly in distinguishing between real and fake news articles. In parallel, Random Forest (RF), an ensemble method composed of 100 decision trees, was trained on the same feature sets to explore the benefits of model averaging and its resilience to overfitting. In addition, Multinomial Naive Bayes (NB) was applied, albeit exclusively on TF-IDF features, to capitalize on its simplicity and strong performance on text data with discrete distributions.

Each classifier was trained and evaluated using an 80/20 split of the dataset into training and test sets. Performance was assessed through standard classification metrics, including accuracy, precision, recall, and F1-score. To further elucidate model behavior, confusion matrices were constructed and visualized, providing insight into the distribution of true and false predictions across classes.

3.3 Deep Learning Models

In addition to traditional machine learning approaches, we implemented deep learning models to capture sequential dependencies in textual data. Specifically, we developed two neural network architectures: a Simple Recurrent Neural Network (RNN) and a Long Short-Term Memory (LSTM) network.

Both models utilize an embedding layer to convert input text into dense vector representations. The vocabulary size was determined from the training corpus, and the embedding dimension was set to 40. The input sequences were padded to a uniform length to ensure compatibility with the fixed input size expected by the models.

The RNN model consists of an embedding layer, followed by a SimpleRNN layer with 100 hidden units, and a final dense layer with a sigmoid activation function for binary classification. The model was trained using the Adam optimizer with binary cross-entropy as the loss function. Training was conducted over 20 epochs with a batch size of 32. Model performance was evaluated using accuracy and loss metrics on both training and validation datasets, with results visualized over epochs.

The LSTM model follows a similar architecture but replaces the recurrent layer with an LSTM layer comprising 100 units. To mitigate overfitting, dropout layers with a rate of 0.3 were applied after the embedding and LSTM layers. The remaining configuration, including the optimizer, loss function, and training parameters, mirrors that of the RNN model. Evaluation metrics for the LSTM model also included accuracy, precision, recall, and F1-score, along with graphical analysis of training dynamics.

Confusion matrices for both models were generated to further analyze classification performance, highlighting the models' ability to differentiate between fake and real news articles.

3.4 Transformer-based Model

To leverage state-of-the-art language understanding capabilities, we employed a transformer-based model, specifically BERT (Bidirectional Encoder Representations from Transformers). The pre-trained bert-base-uncased model from the HuggingFace Transformers library was fine-tuned on our fake news dataset.

Input sequences, constructed by concatenating the title and author fields, were tokenized using the BERT tokenizer with padding and truncation applied to a maximum sequence length of 512 tokens. Tokenized outputs included input_ids and attention_mask, both formatted as PyTorch tensors. A custom dataset class was implemented to integrate these tokenized inputs with their corresponding labels, enabling efficient batch processing via PyTorch's DataLoader.

The fine-tuning process utilized a BERT model for sequence classification with two output classes. Model training was conducted over six epochs with a batch size of 16. The AdamW optimizer was employed with a learning rate of 5×10^{-5}. During each epoch, training loss was monitored and averaged, while model evaluation was performed on a held-out test set. Evaluation metrics included accuracy, precision, recall, and F1-score, supplemented by confusion matrix analysis.

Training dynamics were further visualized through plots of training loss and test accuracy across epochs, providing insights into the model's convergence behavior and generalization performance.

4. Dataset

To train and evaluate our fake news detection models, we utilized a publicly available dataset from Kaggle, which is widely used in binary news classification research. The dataset consists of 20,800 news articles, each annotated as real or fake. It includes five attributes: *id*, *title*, *author*, *text*, and *label*. The label field indicates whether an article is *real* (0) or *fake* (1).

For all modeling approaches, the title and author fields were concatenated to form the primary textual input. This combined field served as the basis for feature extraction and tokenization across traditional machine learning, deep learning, and transformer-based models. The text field, while available, was not used directly due to its variability in length and content.

The dataset is relatively balanced, providing an equitable distribution between real and fake news, which supports reliable model evaluation (see **Figure 2**).

4.1 Dataset Statistics

Prior to training, standard preprocessing steps were applied, including handling of missing values and text normalization. Missing values in the *title* and *author* fields were replaced with empty strings to maintain data consistency. The dataset was then split into training and test sets using an 80:20 ratio. A summary of the dataset's structure is presented in **Table 1**.

Table 1: Dataset Attributes and Completeness Overview

Attribute	Non-null Count	Data Type	Description
id	20,800	Integer	Unique identifier for each news article
title	20,242	String	The title of the news article

Table 1: Dataset Attributes and Completeness Overview			
author	18,843	String	Author of the article (may contain missing values)
text	20,761	String	Full body of the news content
label	20,800	Integer	Classification label: 1 for fake, 0 for real

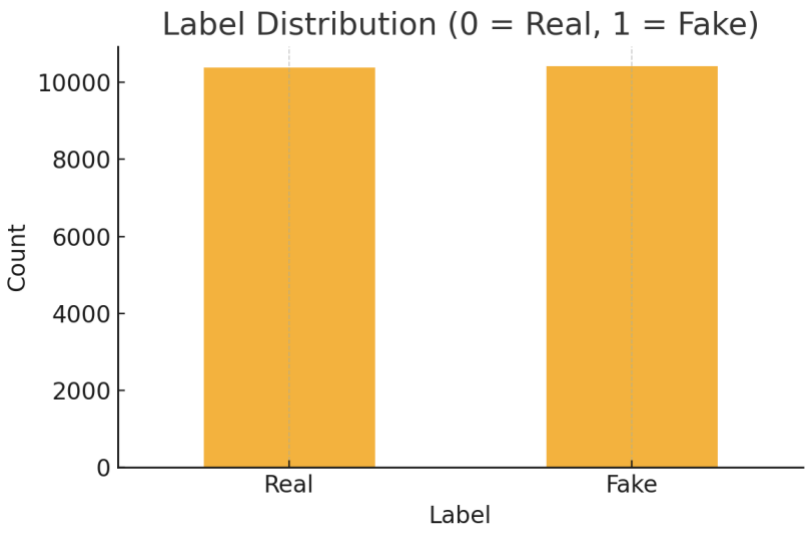


Figure 2: Label Distribution of Real and Fake News in the Dataset

5. Results

In this section, authors evaluated the performance of three traditional machine learning models (SVM, Random Forest, Naive Bayes), two deep learning models (RNN and LSTM), and one transformer-based model (BERT) on the task of fake news classification. All models are tested on preprocessed datasets using consistent evaluation metrics, including accuracy, precision, recall, and F1-score. The following sections compare the results within and across different models and feature representations.

5.1 Machine Learning Models Results

Feature extraction is performed using two textual representation techniques: TF-IDF and Bag-of-Words (BoW). These transformed features are then input into three traditional machine learning classifiers—Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB)—to evaluate their predictive performance. Each model is trained separately on both types of feature vectors, resulting in six configurations.

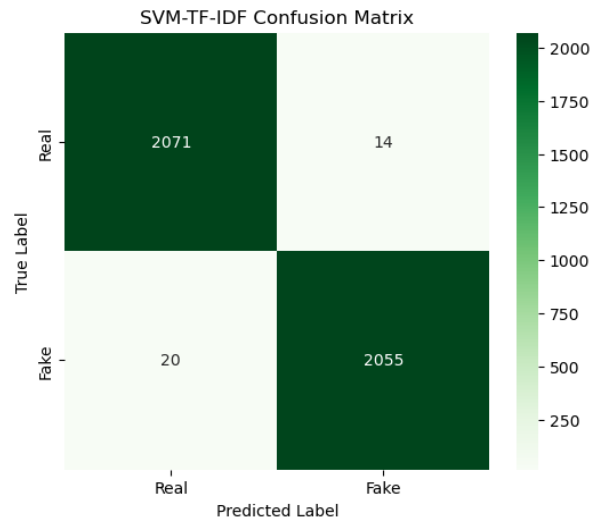
The results are shown in **Table 2** where the six mentioned models' performance evaluation metrics are demonstrated. All models achieved high performance, with Random Forest using TF-IDF achieving the highest accuracy of 99.35%. Both SVM and RF models consistently demonstrated strong precision and F1-scores across vectorization strategies. Naive Bayes, while slightly lower in overall accuracy (95.38% with TF-IDF and 96.23% with BoW), maintained reasonably high precision, indicating that its false positive rate was controlled.

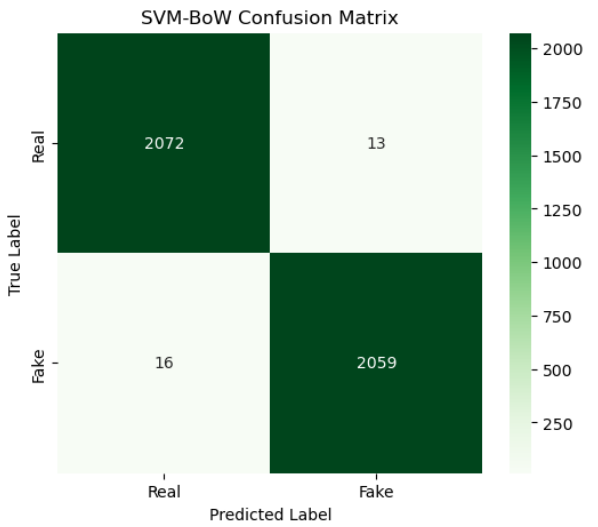
Table 2: Classification results of ML models with textual features

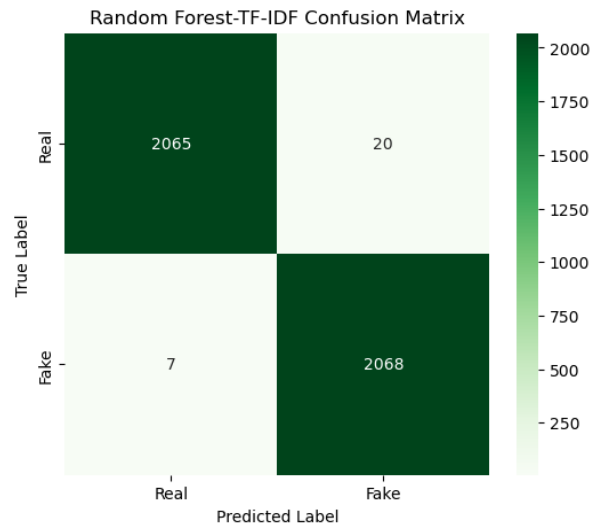
PEM	SVM-TF-IDF (%)	SVM-BoW (%)	RF-TF-IDF (%)	RF-BoW (%)	NB-TF-IDF (%)	NB-BoW (%)
Accuracy	99.18	99.30	99.35	99.28	95.38	96.23
Precision	99	99	99	99	96	96
F1-score	99	99	99	99	95	96
Recall	99	99	99	99	95	96

To better understand model behavior beyond aggregate metrics, confusion matrices were analyzed (see **Figure 3**). For both SVM and RF, false positives and false negatives were relatively balanced, with only minor misclassifications. However, the Naive Bayes model exhibited a significantly higher number of false negatives, particularly when using TF-IDF features. This suggests a tendency to underpredict fake news instances, possibly due to its assumption of feature independence, which is less robust in the presence of complex linguistic patterns.

Figure 3: Confusion matrix comparison of ML models with textual features







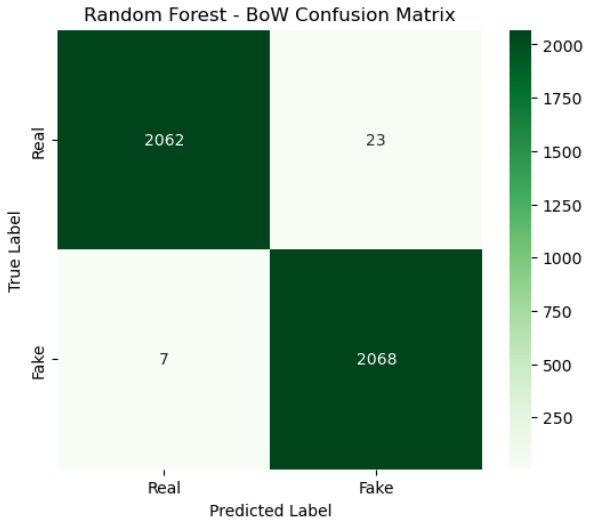
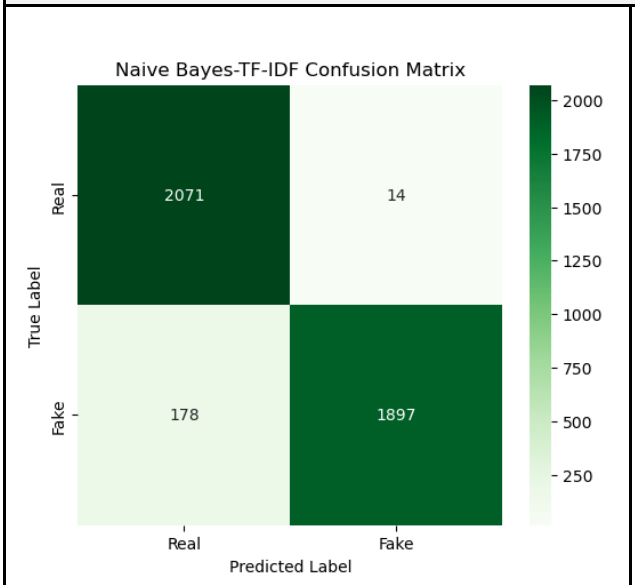
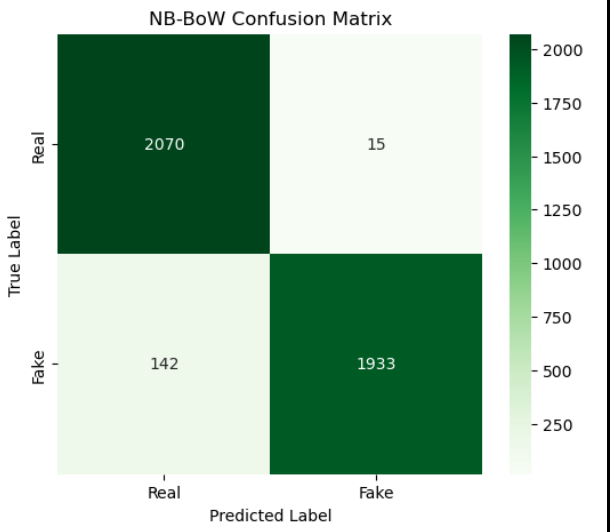


Figure 3: Confusion matrix comparison of ML models with textual features





Error Analysis:

To further understand the sources of misclassification, classification errors are analyzed by examining the most frequent tokens in incorrectly predicted samples, distinguishing between false positives (Real misclassified as Fake) and false negatives (Fake misclassified as Real). Terms such as *Trump*, *Obama*, *New York*, *Time*, and *Report* were repeatedly associated with both false positives and false negatives across different models. A summary of misclassification patterns is shown in **Table 3**.

Table 3: Thematic Patterns in Misclassifications Across ML Models

Misclassification Theme	Common Topics/Terms	Possible Cause
Political Figures	Trump, Clinton, Obama, Israel	Politically charged terms appear in both fake and real news, making them hard to disambiguate.
Journalistic Style	New, York, Time, Report, News	Words commonly used in journalistic writing create a false sense of credibility, leading to fake news being mislabeled as real.
Partisan Language	Breitbart, Impeach, Demand, Ban	Strong rhetoric triggers misclassification as fake.
Ambiguous Entities	John, J, Xenaki, Edjenn	Presence of ambiguous or lesser-known names may increase model uncertainty, especially if context is limited.

These patterns suggest that models sometimes conflate legitimacy with tone or named entities. Articles using formal journalistic language or referencing public figures are occasionally misclassified due to the models' over-reliance on surface-level linguistic cues. Additionally, the

presence of uncommon names or out-of-vocabulary terms may lead to unpredictable classification outcomes due to limited contextual understanding.

In summary, although all machine learning models demonstrate high accuracy in classifying fake news based on textual features, subtle differences in language, especially within political or journalistic content, still remain a challenge for precise classification.

5.2 Deep Learning Models Results

Stemming and word-embedding were performed using keras. Then two models are trained using the results: RNN and LSTM. Both models were evaluated on the test set using standard classification metrics. The performance of these models was not good on the given dataset. It shows some trend of overfitting.

The RNN starts with an Embedding layer that transforms word indices into dense vector representations, followed by a SimpleRNN layer with 100 units to capture temporal patterns in the sequence. A Dropout layer with a rate of 0.5 is applied to reduce overfitting. The final Dense layer uses a sigmoid activation function to output a probability for binary classification. The model is compiled with the Adam optimizer, binary cross-entropy loss, and accuracy as a performance metric. During training, an EarlyStopping callback monitors validation loss and halts training if no improvement is seen over three consecutive epochs, restoring the best weights.

The LSTM is similar but has a dropout layer with a rate of 0.3. Both of the models were trained for 20 epochs to seek the best performance. Needs to be mentioned, that both of the models were trained without the callback monitor at first. But after 20 epochs, the loss on the test set increases significantly and reaches 0.5 when accuracy is also high. This means the model does make the right prediction, but the prediction is always close to 0.5, which is a clear signal of overfitting. So a callback monitor was introduced to reduce this effect. The final results are below.

The final evaluation results, as shown in **Table 4**, demonstrate that the LSTM model slightly outperforms the RNN model in terms of accuracy, achieving 92.98% compared to the RNN's 92.79%. In terms of precision, recall, and F1-score, both models achieved identical values of 93%, indicating similar capabilities in balancing false positives and false negatives. These results suggest that while LSTM has a marginal advantage, likely due to its ability to better model long-term dependencies, the simpler RNN remains nearly as effective in this task. This may imply that the textual patterns within the dataset were not sufficiently complex to necessitate the advanced memory capabilities provided by the LSTM architecture.

Table 4: Classification results of DL models with textual features

PEM	RNN	LSTM
Accuracy	92.79	92.98
Precision	93	93
F1-score	93	93
Recall	93	93

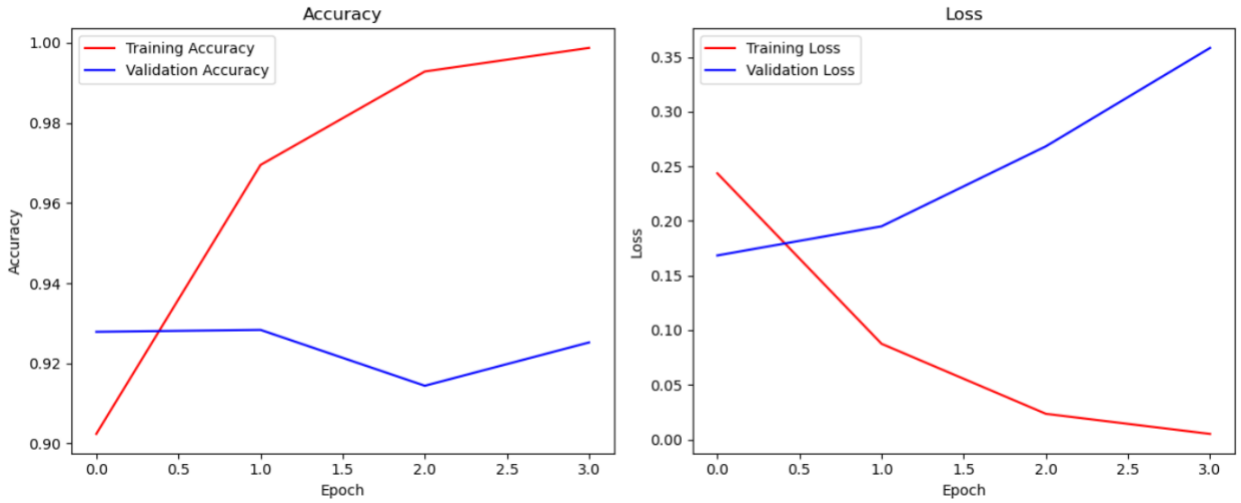


Figure 4: Loss and Accuracy of RNN Across Epochs

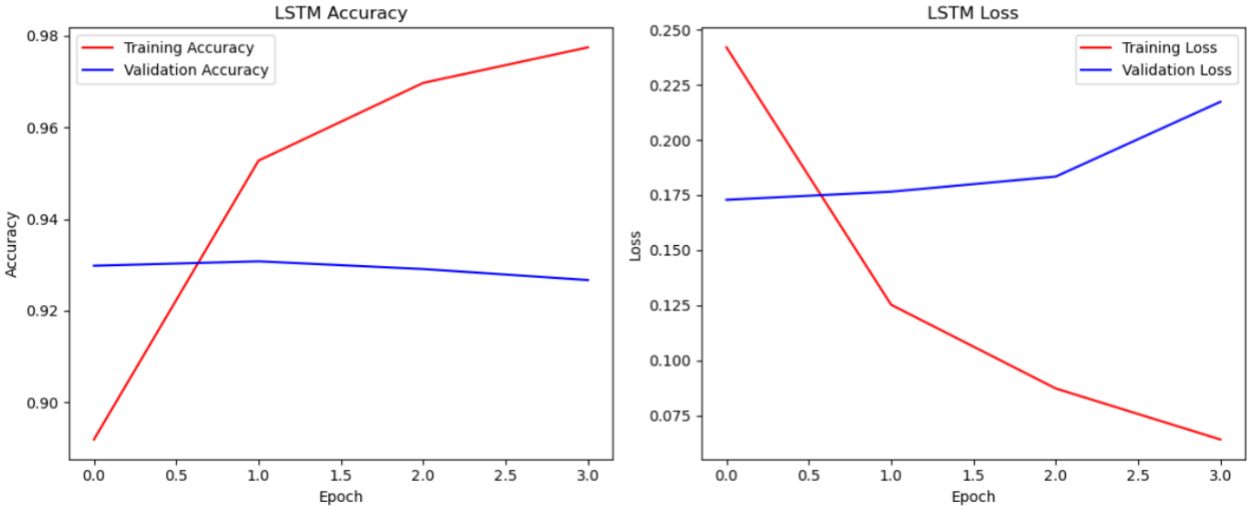
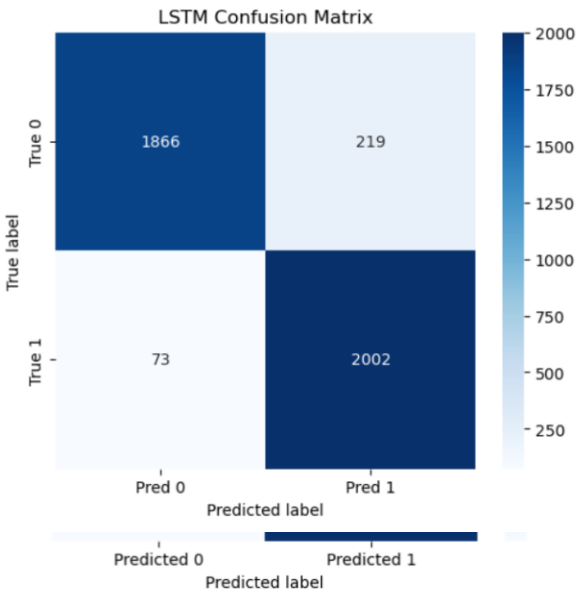


Figure 5: Loss and Accuracy of LSTM Across Epochs

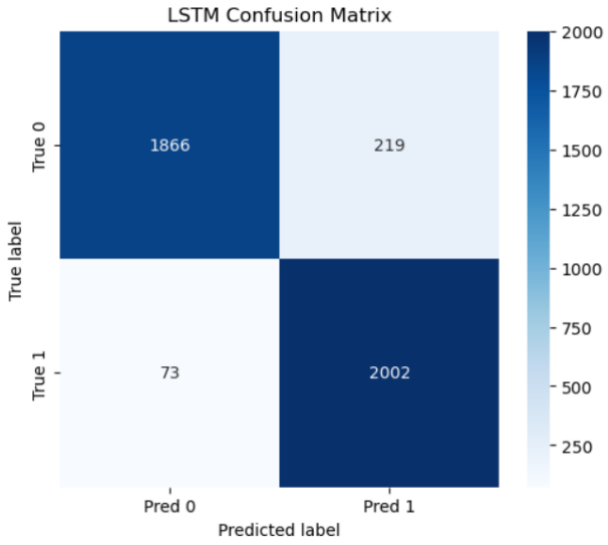
The above training graphs (see **Figure 4**, **Figure 5**) for both RNN and LSTM models show strong performance on the training data, with accuracy increasing and loss decreasing rapidly over epochs. However, validation metrics tell a different story—both models exhibit overfitting, as validation accuracy plateaus or declines while validation loss increases. This suggests that while the models are learning the training data well, they are not generalizing effectively to unseen data.

Compared to RNN, the LSTM shows slightly better generalization but still suffers from the same overfitting trend, indicating a need for stronger regularization or earlier stopping.

Figure 6: Confusion matrix comparison of DL models



Confusion Matrix of RNN



Confusion Matrix of LSTM

The confusion matrices (see **Figure 6**) compare the performance of the RNN and LSTM models on binary classification. For the RNN, the model correctly classified 1843 instances of class 0 and 2017 of class 1, while misclassifying 242 class 0 samples as class 1 (false positives) and 58 class 1 samples as class 0 (false negatives). For the LSTM, the model correctly classified 1866 of class 0 and 2002 of class 1, with 219 false positives and 73 false negatives. This shows that both models perform well, but the LSTM slightly improves true negatives and reduces false positives, though it has a minor increase in false negatives. Overall, LSTM demonstrates slightly better balance and precision in handling both classes.

Error Analysis

Table 5: Thematic Patterns in Misclassifications Across DL Models

Misclassification Theme	**Common Topics/Terms**	**Possible Cause**
Political content	Trump, Hillary Clinton, Pence, election	Model may associate frequently politicized figures with fake news due to biased training data
Sensational language	“Humiliated”, “Hide”, “Jailed”	Emotionally charged or clickbait-like wording mimics typical fake news tone
Reputable sources with dramatic headlines	The New York Times, bold statements	Model relies more on title phrasing than source credibility, leading to misjudgment

Analyzing the examples provided, a clear pattern shows in the model’s false positives (see **Table 5**). Most of the misclassified headlines relate to politically charged or socially sensitive topics, such as Trump-Russia relations, abortion, Hillary Clinton, and Iran. These themes are often associated with misinformation in real-world contexts, which might have biased the model during training. Additionally, some titles contain sensational or emotionally suggestive language, like

“Humiliated Hillary Tries To Hide”, which mimics the tone of fake news headlines and likely influenced the model’s decision. Notably, even credible sources like The New York Times are present in several examples, suggesting the model may not effectively distinguish between reliable and unreliable sources when topic cues are strong. This implies that the deep learning models may be overfitting to topic-related keywords or tone patterns rather than learning deeper contextual signals for news credibility.

5.3 Transformer-based Model Results

The performance of the fine-tuned BERT model was evaluated on the test set using standard classification metrics. The model achieved consistently high accuracy across multiple training epochs, demonstrating its robustness in distinguishing between real and fake news articles.

During training, BERT was fine-tuned over six epochs with a batch size of 16, using the AdamW optimizer with a learning rate of 5×10^{-5}. The progression of training loss and test accuracy across epochs is depicted in **Figure 7**, illustrating a stable convergence pattern.

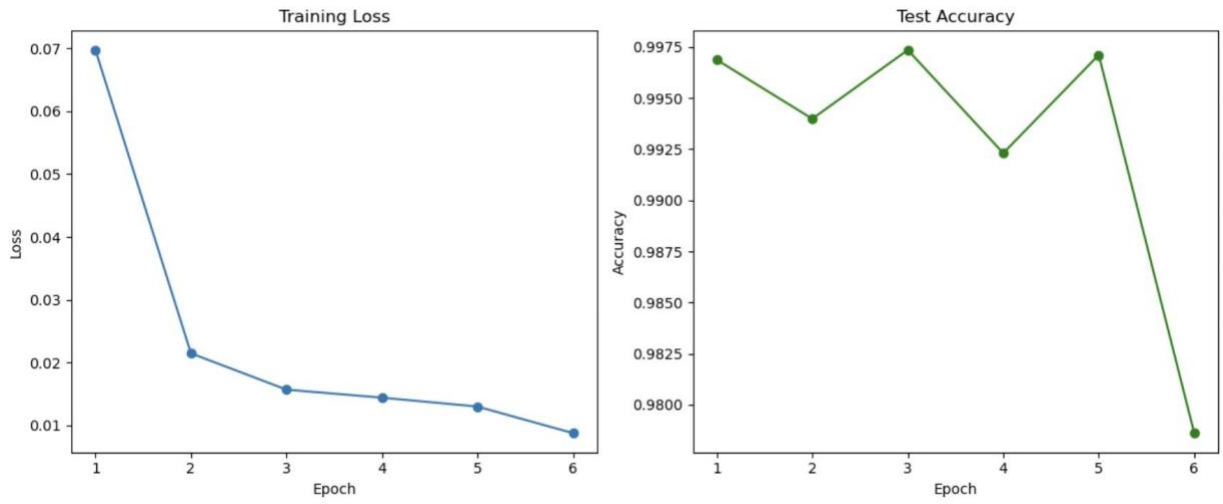


Figure 7: Training Loss and Validation Accuracy of BERT Across Epochs

The evaluation results on the test set are summarized in **Table 6**. The model achieved an overall accuracy of 97.86%, with precision, recall, and F1-score each reaching 98%, indicating a balanced and reliable classification performance.

Table 6: BERT Classification Metrics on Test Set

Metric	**Value (%)**
Accuracy	97.86
Precision	98
Recall	98
F1-Score	98

In addition to these metrics, a confusion matrix was generated to further assess the classification outcomes (**Figure 8**). The matrix reveals minimal false positives and false negatives, underscoring the model's effectiveness.

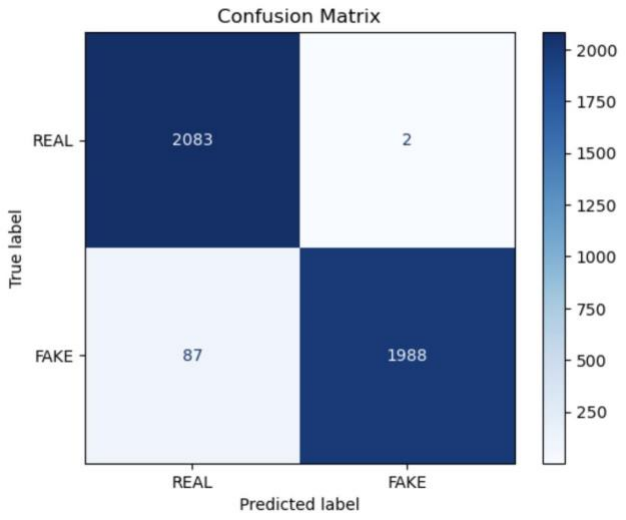


Figure 8: Confusion Matrix of BERT Predictions on the Test Set

Compared to traditional machine learning and deep learning models, BERT demonstrated superior performance, particularly in handling nuanced language representations inherent in fake news detection tasks. This highlights the advantage of transformer-based models in leveraging contextual embeddings over classical word-level representations.

While these quantitative metrics highlight the robustness of the BERT model, a deeper analysis of misclassified samples was undertaken to identify performance limitations and areas for improvement.

Error Analysis

A focused examination of the misclassified instances revealed a noticeable tendency toward false negatives, where fake news articles were incorrectly predicted as real. Among the total misclassifications, the majority involved articles that were labeled FAKE (1) but predicted as REAL (0). This conservative bias suggests that the model requires stronger signals to classify content as fake, potentially relying on surface-level credibility cues.

Further qualitative analysis uncovered recurring patterns in these false negatives (see **Table 7**). A significant proportion of misclassified articles centered on political content, especially those referencing well-known figures such as *Trump*, *Clinton*, and *Obama*. The inclusion of such authoritative names may have biased the model toward real classifications. Another cluster of misclassifications involved health-related claims promoting alternative remedies like *garlic* or *seaweed*. These topics, although common in fake news, may have been underrepresented during training. Additionally, articles using conspiratorial language—mentioning entities like *NSA* or geopolitical issues such as *cash bans*—were frequently misclassified, indicating the model's challenge in distinguishing between legitimate controversy and deliberate misinformation.

Table 7: Summary of Misclassification Patterns in BERT

Misclassification Theme	**Common Topics/Terms**	**Possible Cause**
Political Content	Trump, Clinton, Obama	Authoritative names bias toward real
Health Claims	Garlic, Seaweed, Coconut Oil	Underrepresented in training data

Table 7: Summary of Misclassification Patterns in BERT		
Conspiratorial Language	NSA, Saudi Arabia, Cash Ban	Controversy ≠ deception misinterpretation

The model's reliance on formal language patterns and recognized entities may have overshadowed subtle linguistic cues of deception such as sensationalism or emotional exaggeration. To enhance classification performance, particularly in reducing false negatives, future improvements could include the integration of additional stylistic or sentiment-based features. The application of explainable AI techniques, such as SHAP or LIME, may also provide valuable insights into token-level contributions to predictions, guiding targeted retraining efforts. Moreover, augmenting the training dataset with a broader spectrum of fake news articles, particularly those employing ambiguous language, could further improve the model's generalization capabilities.

6. Discussion

From the previous analyses, it is evident that BERT outperforms both traditional machine learning and deep learning models in the task of fake news detection, particularly in terms of accuracy and robustness. However, model performance is only one side of the equation. When considering real-world deployment, efficiency and cost become equally important factors.

During our training process, we observed significant differences in processing times across model types. Despite limiting the input features to only the title and author, BERT models still required

over 10 minutes per epoch, whereas deep learning models such as RNNs and LSTMs completed an epoch in approximately 10 seconds, and traditional machine learning models executed their tasks within a few seconds. These differences, while manageable at a small experimental scale, could become exponentially more significant in industry-scale applications, where large volumes of streaming data must be processed in real-time or near real-time. Therefore, the 2–3% performance gain from BERT must be carefully weighed against its computational cost and resource requirements, particularly in scenarios where time efficiency and scalability are critical.

6.1 Challenges

Several challenges emerged during this project. First, the dataset was limited to titles and authors, omitting the full article content that could have provided richer semantic information and potentially improved classification accuracy. This restriction likely contributed to the second challenge, overfitting, as deep learning models such as RNNs and LSTMs began to overfit after only a few epochs. The limited input dimensions and lack of deep contextual richness made it difficult for models to generalize. Another observed challenge was the tendency of some models to misclassify real news articles with dramatic phrasing as fake. This suggests that certain models relied more heavily on superficial stylistic cues rather than truly understanding the underlying content. Finally, while BERT demonstrated outstanding classification performance, it imposed substantial computational demands, making the trade-off between model quality and training efficiency a critical consideration for practical deployment.

6.2 Future Work

Several avenues can be explored in future work to enhance both the effectiveness and practicality of fake news detection systems. Incorporating the full text of news articles, along with relevant metadata, would provide models with a richer context and help improve predictive accuracy. Investigating lightweight attention-based hybrid models, or combining deep learning techniques

with rule-based systems, may offer a more favorable balance between accuracy and computational efficiency. Additionally, model optimization techniques such as knowledge distillation, quantization, or pruning could be employed to reduce the runtime and resource consumption of BERT models while retaining most of their performance. Future studies should also focus on real-world evaluation, testing models on live-streamed or multilingual datasets to assess their generalizability and robustness across diverse environments. Finally, enhancing the explainability of model predictions is crucial; building interpretation tools would not only increase user trust but also provide transparency, which is particularly important in sensitive applications like misinformation control.

7. References:

Ahmad, I., Yousaf, M., Yousaf, S., & Ahmad, M. O. (2020). Fake news detection using machine learning ensemble methods. *Complexity*, *2020*(1), 8885861.

Anoop, K., Deepak, P., & Lajish, V. L. (2020, August). Emotion cognizance improves health fake news identification. In *IDEAS* (Vol. 2020, p. 24th).

Bahad, P., Saxena, P., & Kamal, R. (2019). Fake news detection using bi-directional LSTM-recurrent neural network. *Procedia Computer Science*, *165*, 74-82.

Choudhary, M., Chouhan, S. S., Pilli, E. S., & Vipparthi, S. K. (2021). BerConvoNet: A deep learning framework for fake news classification. *Applied Soft Computing*, *110*, 107614.

Di Domenico, G., Sit, J., Ishizaka, A., & Nunan, D. (2021). Fake news, social media and marketing: A systematic review. *Journal of business research*, *124*, 329-341.

Guo, C., Cao, J., Zhang, X., Shu, K., & Liu, H. (2019). Dean: Learning dual emotion for fake news detection on social media (arXiv preprint). *arXiv preprint arXiv:1903.01728*.

Hakak, S., Alazab, M., Khan, S., Gadekallu, T. R., Maddikunta, P. K. R., & Khan, W. Z. (2021). An ensemble machine learning approach through effective feature extraction to classify fake news. *Future Generation Computer Systems*, *117*, 47-58.

Han, Y., Karunasekera, S., & Leckie, C. (2021, September). Continual learning for fake news detection from social media. In *International conference on artificial neural networks* (pp. 372-384). Cham: Springer International Publishing.

Janze, C., & Risius, M. (2017). Automatic detection of fake news on social media platforms.

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia tools and applications*, *80*(8), 11765-11788.

Khan, J. Y., Khondaker, M. T. I., Afroz, S., Uddin, G., & Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, *4*, 100032.

Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017). Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.

Pulido, C. M., Ruiz-Eugenio, L., Redondo-Sama, G., & Villarejo-Carballido, B. (2020). A new

application of social impact in social media for overcoming fake news in health. *International journal of environmental research and public health*, *17*(7), 2430.

Saeed, A., & Al Solami, E. (2023). Fake News Detection Using Machine Learning and Deep Learning Methods. *Computers, Materials & Continua*, *77*(2).

Shaik, M. A., Sree, M. Y., Vyshnavi, S. S., Ganesh, T., Sushmitha, D., & Shreya, N. (2023, March). Fake news detection using NLP. In *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)* (pp. 399-405). IEEE.

Zhu, H., Wu, H., Cao, J., Fu, G., & Li, H. (2018). Information dissemination model for social media with constant updates. *Physica A: Statistical Mechanics and its Applications*, *502*, 469-482.

8. Appendix

8.1 Materials

1) Demo folder: Contains the runnable **demo.py** to execute the demo. Also, have all the saved models and three separate files to execute the three models individually (unit tests for all functions are partially written inside their respective function files, and partially in separate unit test files).
2) Deeplearning_XinyuLiu.ipynb: Contains the trained RNN & LSTM models and unit tests.
3) Machinelearning_XueniTang.ipynb: Contains the trained SVM, RF, and NB models and unit tests.
4) Transformer_JieniTang.ipynb: Contains the trained BERT model and unit tests.
5) Presentation: Contains the presentation slides.
6) Train.csv: Contains the dataset used to train and test the model.

8.2 Required Packages

Install the necessary libraries using pip:

pip install numpy pandas scikit-learn nltk matplotlib

pip install torch torchvision torchaudio

pip install transformers

pip install tqdm

pip install tensorflow keras seaborn

User Manual:

To run the demo.py, open the demo folder in an IDE eg. VSCode, or terminal.

Enter ‘python demo.py’ or click on the run button to execute the file.

Users are asked whether to use pre-defined news or enter their ones. If enter ‘yes’, a piece of pre-defined news will be used. If enter ‘no’, the user will be asked to enter a news title and author.

The models will be used to predict whether the new is real or fake and a result will be presented.

The demo will ask the user to enter ‘yes’ to try a new round or enter ‘no’ to quit.